Claims

WHAT IS CLAIMED IS:

1. A computer program product encoding a computer program for executing on a computer system a computer process for building compact garbage collection tables adapted for use in reclaiming memory from a heap during runtime, the computer process comprising:

generating a first call site table storing call site identifiers;

generating a final descriptor table storing a set of unique descriptors, at least one unique descriptor describing a location of a pointer into the heap; and

generating a descriptor reference table associated with the first call site table, each entry in the descriptor reference table mapping a call site identifier in the first call site table to one of the unique descriptors in the final descriptor table.

2. The computer program product of claim 1 wherein at least two call site identifiers are mapped to the same unique descriptor in the final descriptor table.

3. The computer program product of claim 1 wherein the final descriptor table contains no identical descriptors.

4. The computer program product of claim 1 wherein the operation of generating a first call site table comprises:

storing return addresses for one or more call sites into the call site table.

5. The computer program product of claim 1 wherein the operation of generating a final descriptor table comprises:

generating an initial descriptor table including at least two identical descriptors, each descriptor in the initial descriptor table corresponding with a call site identifier in the call site table;

copying each descriptor from the initial descriptor table to the final descriptor table, if the descriptor is not identical to another descriptor already copied to the final descriptor table.

6. The computer program product of claim 1 wherein the operation of generating a descriptor reference table comprises:

generating a table pair including a second call site table and an initial descriptor table, the initial descriptor table storing descriptors that include at least two identical descriptors;

sorting the table pair based on the descriptors in the initial descriptor table to provide a sorted table pair;

traversing sequentially through the descriptors in the sorted table pair to associate a reference to each call site in the second call site table, the reference being modified when a unique descriptor is encountered in the initial descriptor table;

identifying each call site identifier in the second call site table to which each reference is associated; and

storing each reference into the descriptor reference table in association with the call site identifier identified in the identifying operation.

7. The computer program product of claim 6 wherein the reference is an ordinal identifier of unique descriptors being processed during the traversing operation.

24

8. The computer program product of claim 7 wherein the traversing operation comprises:

incrementing the ordinal identifier when a unique descriptor is encountered in the initial descriptor table.

9. The computer program product of claim 6 wherein the reference includes a pointer to one of the unique descriptors stored in the final descriptor table.

10. The computer program product of claim 9 wherein the traversing operation comprises:

designating as the reference a new pointer to one of the unique descriptors in the final descriptor table when a unique descriptor is encountered in the initial descriptor table.

11. A computer-readable medium having stored thereon compact garbage collection tables for identifying elements of a root set used in reclaiming memory from a heap during runtime, the compact garbage collection tables comprising:

a call site table storing call site identifiers;

a final descriptor table storing a set of unique descriptors, each unique descriptor describing a location of a pointer into the heap, at least one descriptor identifying an element of the root set; and

a descriptor reference table associated with the call site table, each entry in the descriptor reference table mapping a call site identifier in the call site table to a unique descriptor in the descriptor table.

12. The computer readable medium of claim 11 wherein at least two call site identifiers of the call site table are mapped to the same unique descriptor in the descriptor table.

13. The computer readable medium of claim 11 wherein the descriptor table contains no identical descriptors.

14. A method of building compact garbage collection tables adapted for use in reclaiming memory from a heap during runtime, the method comprising:

generating a first call site table storing call site identifiers;

generating a final descriptor table storing a set of unique descriptors, at least one unique descriptor describing a location of a pointer into the heap; and

generating a descriptor reference table associated with the first call site table, each entry in the descriptor reference table mapping a call site identifier in the first call site table to one of the unique descriptors in the final descriptor table.

15. The method of claim 14 wherein at least two call site identifiers of the first call site table are mapped to the same unique descriptor in the final descriptor table.

16. The method of claim 14 wherein the final descriptor table contains no identical descriptors.

17. The method of claim 14 wherein the operation of generating a first call site table comprises:

storing return addresses for one or more call sites into the call site table.

18. The method of claim 14 wherein the operation of generating a final descriptor table comprises:

generating an initial descriptor table including at least two identical descriptors, each descriptor in the initial descriptor table corresponding with a call site identifier in the call site table; and

27

copying each descriptor from the initial descriptor table to the final descriptor table, if the descriptor is not identical to another descriptor already copied to the final descriptor table.

19. The method of claim 14 wherein the operation of generating a descriptor reference table comprises:

generating a table pair including a second call site table and an initial descriptor table, the initial descriptor table storing descriptors that include at least two identical descriptors;

sorting the table pair based on the descriptors in the initial descriptor table to provide a sorted table pair;

traversing sequentially through the descriptors in the sorted table pair to associate a reference to each call site in the second call site table, the reference being modified when a unique descriptor is encountered in the initial descriptor table;

identifying each call site identifier in the second call site table to which each reference is associated; and

storing each reference into the descriptor reference table in association with the call site identifier identified in the identifying operation.

20. The method of claim 19 wherein the reference is an ordinal identifier of the descriptor and call site identifier being processed during the traversal of the sorted table pair.

21. The method of claim 20 wherein the traversing operation comprises:

incrementing the ordinal identifier when a unique descriptor is encountered in the initial descriptor table.

28

22. The method of claim 19 wherein the reference includes a pointer to one of the unique descriptors stored in the final descriptor table.

23. The method of claim 22 wherein the traversing operation comprises:

designating as the reference a new pointer to one of the unique descriptors in the final descriptor table when a unique descriptor is encountered in the initial descriptor table.

24. A computer program product encoding a computer program for executing on a computer system a computer process for identifying elements of a root set for garbage collection using compact garbage collection tables, the computer process comprising:

locating a call site identifier in a call site table;

identifying a descriptor reference in a descriptor reference table, the descriptor reference being associated with the call site identifier;

identifying a descriptor referenced by the descriptor reference, the descriptor reference mapping between the call site identifier and the descriptor, the descriptor being one descriptor of a set of unique descriptors in a descriptor table; and

accessing the descriptor to determine the elements of the root set for garbage collection.

25. The computer program product of claim 24 wherein the descriptor is mapped to at least two call site identifiers in the call site table.

26. The computer program product of claim 24 wherein the descriptor table contains no identical descriptors.

27. The computer program product of claim 24 wherein the descriptor reference includes an index into the descriptor table.

28. The computer program product of claim 24 wherein the descriptor reference includes a pointer into the descriptor table.

29. A runtime system for identifying elements of a root set for a garbage collection using compact garbage collection tables, the runtime system comprising:

a garbage collector accessing a call site table storing call site identifiers;

a descriptor table storing a set of unique descriptors, each unique descriptor describing a location of a pointer into the heap, and

a descriptor reference table associated with the call site table to identify elements of a root set, each entry in the descriptor reference table mapping one of the call site identifiers in the call site table to a descriptor in the descriptor table, at least one descriptor in the descriptor table being mapped to a plurality of call site identifiers.

30. The runtime system of claim 29 wherein the descriptor table contains no identical descriptors.

31. A method of identifying elements of a root set for garbage collection using compact garbage collection tables, the method comprising:

locating a call site identifier in a call site table;

identifying a descriptor reference in a descriptor reference table, the descriptor reference being associated with the call site identifier;

identifying a descriptor referenced by the descriptor reference, the descriptor reference mapping between the call site identifier and the descriptor, the descriptor being one descriptor of a set of unique descriptors in a descriptor table; and

accessing the descriptor to determine the elements of the root set for garbage collection.

32. The method of claim 31 wherein the descriptor is mapped to at least two call site identifiers in the call site table.

33. The method of claim 31 wherein the descriptor table contains no identical descriptors.